An Innovative Knowledge-Based System Using Fuzzy Cognitive Maps for Command and Control

Final Report

Contract # N66001-97-C-7003
Item # 0001AC
CDRL # A003

Contract Period: 8 May 1997 through 07 November 1997

Sponsored by:
U.S. MARINE CORPS SYSTEMS COMMAND
QUANTICO, VIRGINIA

Administered by:
SPAWARSYSCEN SAN DIEGO, CALIFORNIA

Submitted by:
TACAN Corporation
2330 Faraday Avenue
Carlsbad, CA 92008
(760) 438-1010

Principal Investigator: James H. Bechtel, Ext. 3278

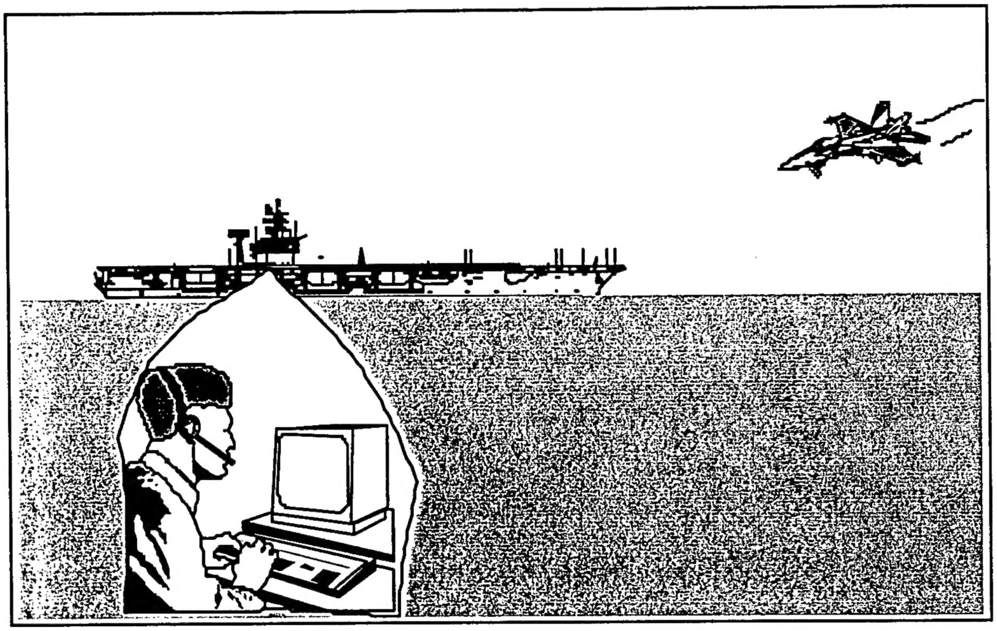
CONTENTS

FIGURES

TABLES

I. INTRODUCTION

The objective of this research program was to explore and investigate innovative concepts and algorithms that can be used to develop a user friendly knowledge-based system for command and control decision making process. Figure 1 shows an example of the possible application of the proposed innovative information processing and data reasoning technology on the task of enhancing decision making process in amphibious warfare.



"Alert! FCM command and control software indicates possible crash status caused by too deep descent trajectory! Abort attempt and circle carrier for new approach vector!"

Figure 1. An Application Example of the Proposed Information Processing Technology.

For an information fusion and data reasoning system, the data from different individual data channels are usually noisy, uncertain, partial, occasionally incorrect and usually incoherent. The data patterns obtained are usually of different sizes and resolution, distorted, rotated and presented in different scales. Information from multiple sensors allows features in the environment to be perceived that are impossible to perceive using just the information from each individual sensor operating separately. The information from multiple sensors should be processed in a way to enhance recognition, identifying capabilities and thus the reliability of the decision making process. The processing of redundant information can reduce overall uncertainty, and thus, increase the accuracy with which the features are perceived by the system.

Further, processing of complementary information usually increases the discrimination ability of the information system.

With standard techniques used for the information fusion and data reasoning in the command and control process make explicit assumptions concerning the nature of the sensory information. The most common assumptions include the use of a measurement model for each sensor that includes a statistically independent additive Gaussian error or noise term and an assumption of statistical independence between the error terms for each sensor. Many of the differences in the information fusion and data reasoning methods center on their particular techniques (e.g., calibration, thresholding) for transforming raw sensory data into a form so that the above assumptions become reasonable, and thus a mathematically tractable information fusion and data reasoning method can result. The information fusion and data reasoning methods based on these assumptions are [1]:

1. Kalman Filter [2] [3],
2. Bayesian Estimation using Consensus Sensors [4] [5],
3. Multi-Bayesian [6] [7], and
4. Statistical Decision Theory [8] [9].

A challenge of these statistical classifiers is that the statistical data distributions of the sensor data are usually not exactly known, and the common assumption of Gaussian distribution is not usually valid. This absence of knowledge of the prior probabilities has been long recognized as a difficulty for the Bayesian approach.

Many different artificial neural networks (ANNs) have been applied to the problems of information fusion and data reasoning. As compared to the traditional information fusion and data reasoning methods, such as k nearest-neighbor classifier, Bayesian classifier using Parzen windows, statistic quadratic, rules and weighted sums, it had been demonstrated that the performance of the ANNs is either equivalent or better [10]-[13]. Furthermore, ANNs tend to use far fewer free parameters, and requires much less computations. However, the performance of the ANNs is usually dependent on the following issues:

1. The form of the ANNs, such as the input and the output signals of the ANNs, the architecture of the ANNs, and the learning rules used for learning.
2. The structure of the training and testing data sets used to train and test the ANNs, since the ANNs must be trained to be general and robust enough so that all the possible system scenarios can be handled.

In a practical information fusion and data reasoning system, the information received purely from sensors is usually constrained, and thus, the design and the training of the ANNs are constrained, and therefore, the performance of the system is bounded. Further, due to the use of the sensors' raw data or features data, the classifications made by the ANNs are the low level classifications which usually do not have good discrimination between various pattern types. Furthermore, redesign and retraining of a well trained ANN is usually necessary for the case that more relevant and important information is available. That is, time consuming processes are usually required to "expand" an ANNs based information fusion and data reasoning system.

With the uncertainties and the problems stated above, an information fusion and data reasoning system for command and control, which can be employed for reliable and robust decision making process should be:

1. Intelligent enough to accommodate or even compensate the uncertainties that come from the various inputs.
2. Flexible enough to work with various types of information or sensor data.
3. Adaptable enough to synthesize previous experiences and the knowledge of different domain experts.
4. Friendly enough to let the user edit, create and manipulate the algorithms used for data reasoning and the decision making process.

To fulfill these requirements, based on the concept of Fuzzy Cognitive Map (FCM) [14]-[27], we proposed a research program to investigate and to develop a user friendly knowledge-based system for command and control decision making processes.

A fuzzy cognitive map is a nonlinear feedback dynamical system for modeling causal knowledge process. It is an alternative to conventional expert systems in situations where the available

knowledge has some uncertainties and where the knowledge base resides in a large number of experts, possibly having different levels of expertise and diverse or even conflicting views. As compared to conventional expert systems, fuzzy cognitive maps have the following special characteristics which make them outstanding candidates for use in complex control and planning systems:

1. Inferencing in the expert system involves tree searching and is, therefore, inherently a sequence process. The fuzzy cognitive map allows synchronous updating of all nodes and links (i.e., high speed parallel processing).
2. If the basic structure of the expert system is a tree, two or more expert systems cannot be readily combined. Fuzzy cognitive maps permit, in a simple manner, the combining of knowledge acquired from different sources.
3. Conventional expert systems cannot be adaptively refined through the learning process. The fuzzy cognitive map is capable of adaptive refinement through supervised and unsupervised learning.
4. Knowledge acquisition and its translation into a form suitable for expert system use requires special skills. A domain expert can sketch a fuzzy cognitive map after a very short period of instruction.

The fuzzy cognitive map was proposed by Kosko [14] [15] as an alternative to conventional expert systems in situations where the available knowledge has some uncertainties and where the knowledge base resides in a large number of experts, possibly having different levels of expertise and diverse or even conflicting views. The structure of the fuzzy cognitive map is such that two or more maps can be readily combined, even if the maps differ substantially. When combining different maps acquired from a group of experts, the individual maps may be weighted so as to reflect the level of credibility of each expert. Gathering and combining knowledge from a large number of sources leads to a dynamic structure which preferentially exhibits behavior corresponding to that knowledge which is most firmly known. The acquisition of fuzzy cognitive maps can be accomplished in three ways:

1. by asking for the knowledge from an expert or group of experts.
2. by abstracting situation-response prototypes from historical data.

3. by autonomously growing them in adaptive learning networks.

All three of these methods are likely to be of utility in a planning and control system. Moreover, the methods of acquisition can be freely mixed. For example, a fuzzy cognitive map acquired initially by querying experts might later be refined adaptively during actual use.

Unlike conventional expert systems, fuzzy cognitive maps operate with dynamic feedback. Dynamic feedback permits the use of knowledge which contains some uncertainties. For example, in a fuzzy cognitive map, a logical inference once made is not immutable, but may be changed in light of subsequent inferences. In other words, an inference in a chain of inferences can be altered if it leads to conclusions which are incompatible. In this way a piece of uncertain knowledge can be used as a guide in reaching decisions without locking the system into rigid, and possibly paralyzing, behavior.

In many complex control systems, the human operator participates critically in providing decisions based on experience in situations which are not easily quantified to allow automatic decision making. One may argue that the human is nevertheless following a set of rules and that, in principle, the rules could be written down and followed by an automatic decision maker. Even if such a set of rules is obtained, however, the rule designer is faced with the nearly insurmountable task of verifying that the rules lead to an acceptable decision in all conceivable circumstances, including those which have an infinitesimal probability of occurring. The near impossibility of anticipating all possible consequences of strictly following a set of rules suggests that the rules in a rule-based control system should not be inviolable, but should have provisions for being overridden by a higher authority. The fuzzy cognitive map with its feedback mechanism provides a natural means by which decisions can be overridden. In fact, any inference can be overridden selectively, and the fuzzy cognitive map will immediately converge to "the next best" inference. This allows a human operator, for example, to veto a fuzzy cognitive map decision, based on the human's broader experience and values. In such cases, the fuzzy cognitive map converges to a new solution, automatically overriding first those rules in the knowledge base which are less firmly held.

A primary function of the human operator in many complex control systems is the rapid selection of relevant observational data from the huge quantity of data being gathered and presented. The selection of the relevant data requires much more than a simple filtering of the

data stream since data which is not relevant, in most situations, may suddenly become of critical importance as the situation changes. The operator must also be capable of making decisions in situations which have never arisen before; hence he must have, or develop through experience, an ability to make accurate predictions of future system states given a previously unseen system state. It is in the areas of prediction and data selection that the fuzzy cognitive map is particularly promising for taking some of the burden off the human operator.

II. RESULTS

Task 1: Explore the inherent opportunities of fuzzy cognitive maps.

Approach: This section provides a theoretical basis for fuzzy inferencing in the fuzzy cognitive map.

Structure of the Fuzzy Cognitive Map

The fuzzy cognitive map is a directed graph in which the nodes represent concepts, while the edges represent causal links between concepts. Thus edge w_{ij} represents the causal connection between concept C_i and concept C_j, that is, w_{ij} is a measure of the degree to which the presence of C_j, causes an increase or decrease in C_j. The possible values of w_{ij} lie in the interval [-1, +1]. A value of +1 indicates a maximum causal increase while a value of -1 indicates a maximum causal decrease. The nodes of the fuzzy cognitive map take on values in the range [0,1]. A value of 1 for C_j, indicates the concept represented by C_j, is present to the maximum degree.

Figure 2 illustrates a very simple fuzzy cognitive map: a partial cognitive map for piloting an aircraft. The fuzzy cognitive map in this case represents the causal interconnection between various flight parameters. Pitching the nose of the aircraft up, for example, increases the angle of attack of the wing, which increases lift. Drag is also increased, however, and this tends to reduce airspeed, which then decreases lift. Air speed may be restored, however, by advancing the throttle.

The cognitive map of Figure 2, therefore, represents in a very simple way some of the knowledge that is required to control an aircraft. It is a qualitative causal model. It can be argued that it is cognitive maps of this sort that enable human pilots to quickly adapt their flying skills to aircraft they haven't flown before.

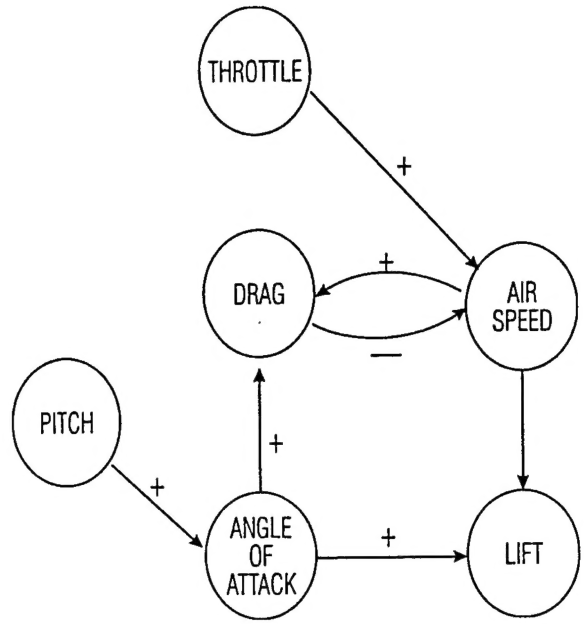


Figure 2. A partial cognitive map for piloting an aircraft.

Combining Fuzzy Cognitive Maps

Although fuzzy cognitive maps obtained from different experts over the same knowledge domain will, in general, have different causal connection strengths and may even utilize different concepts, this presents no great obstacle to combining them in a single map. One simply converts each fuzzy cognitive map to matrix form and adds the matrices together, element by element. If a concept present in one fuzzy cognitive map is not present in another, one simply augments the matrix of the deficient fuzzy cognitive map, filling in the extra rows and columns with zeros.

Figure 3 represents four hypothetical cognitive maps obtained from four experts. Experts A and B used the same concepts in their cognitive maps but disagreed as to some of the causal connections. Expert C utilized a different concept (concept 5 instead of concept 3), while expert D used yet another concept (concept 6 instead of concept 4).

Figure 4 shows the matrix representations of these four cognitive maps, each augmented appropriately so that they may be added together directly. After standard matrix addition of these four matrices, a new matrix representation can be obtained, and Figure 4 shows the resulting composite matrix, normalized so that maximum causality is still represented by a connection strength of one. Finally, with this new matrix, a fuzzy cognitive map which

represents the combination of four experts' knowledge can be generated, and Figure 5 shows the schematic representation of the final composite fuzzy cognitive map.

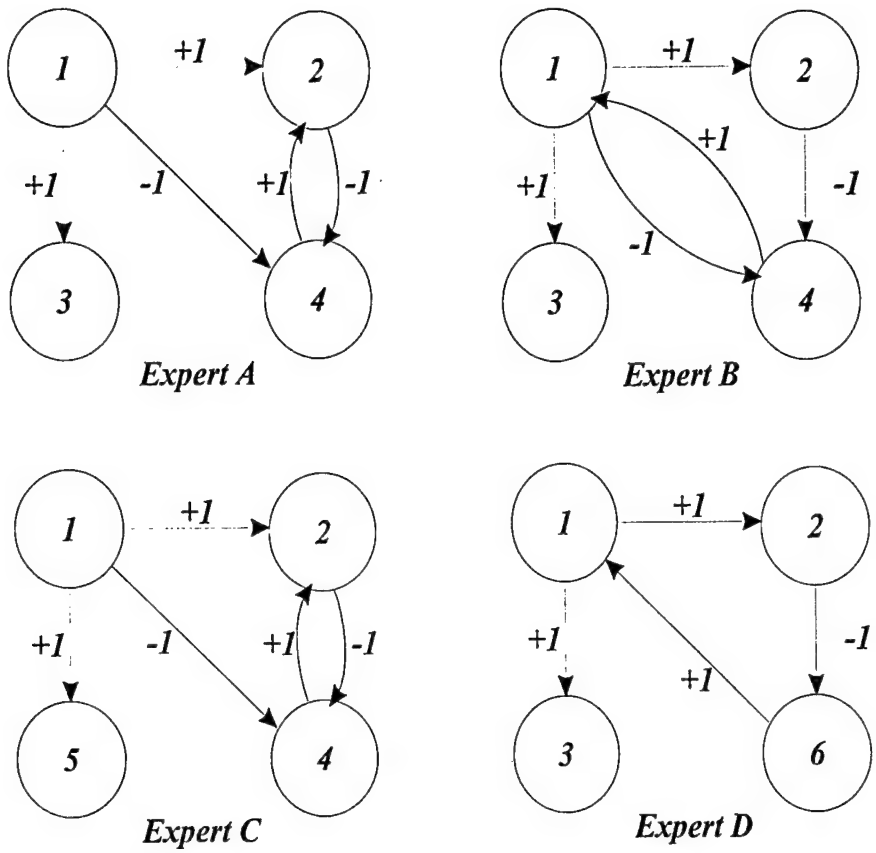


Figure 3. Hypothetical cognitive maps obtained from four different experts.

In the example illustrated above in Figure 3, the fuzzy cognitive map matrices were added together with equal weights. It is also possible to weight each fuzzy cognitive map individually, depending on the credibility or degree of expertise of the source from which it was obtained. In combining several fuzzy cognitive maps, causal connections (edges) differing in sign will tend to cancel. Thus causal connections over which there is some disagreement among experts will be diminished in the composite fuzzy cognitive map. As the knowledge of more experts is added, the composite fuzzy cognitive map will come to represent that knowledge which is most firmly known.

Expert A

$$\begin{bmatrix} 0 & 1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Expert B

$$\begin{bmatrix} 0 & 1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Expert C

$$\begin{bmatrix} 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Expert D

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 4. Matrix representations of the four cognitive maps shown in Figure 3.

$$\begin{bmatrix} 0 & 1 & 0.75 & -0.75 & 0 & 0 \\ 0 & 0 & 0 & -0.75 & 0 & -0.25 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0.25 & 0.25 & 0 & 0 & 0.25 & 0 \\ 0.25 & 0 & 0 & 0 & 0 & 0 \\ 0.25 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 5. Matrix representation of the composite fuzzy cognitive map.

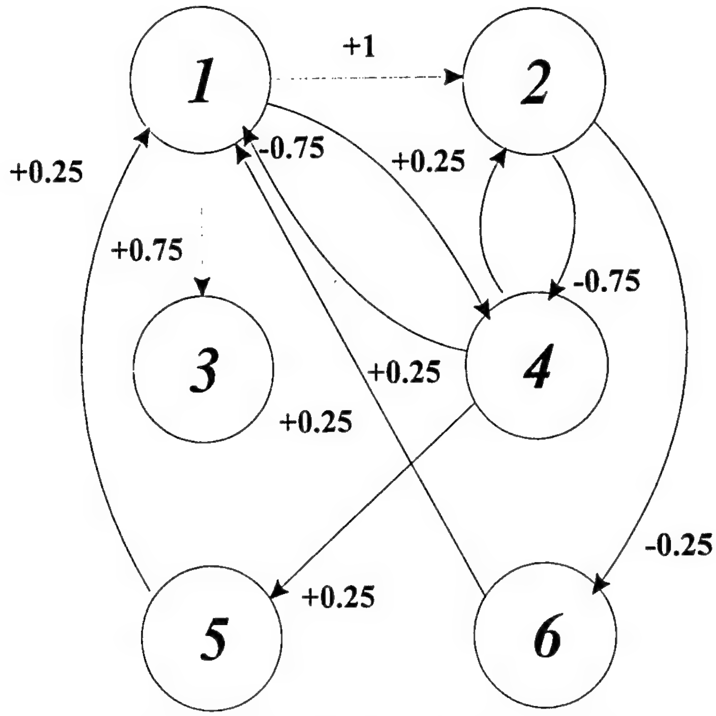


Figure 6. Composite fuzzy cognitive map.

Fuzzy Cognitive Map Application

The fuzzy cognitive map offers a very important application related property which is not fully addressed by other knowledge representation methods: it can be utilized to perform qualitative causal modeling, that is, emulate behavior of a system through qualitative causal modeling. The system may be one for which it is infeasible or undesirable to produce a full quantitative model because:

1. The system is too complex (too many variables or too many interactions) to be modeled in detail in a reasonable amount of time.
2. Portions of the system are not understood well enough to define a quantitative model.
3. Although the system is understood, the data required to build a quantitative model are not available and cannot be easily obtained.
4. The system is subject to unpredictable variation which would render any quantitative model inexact.

5. A generic model is desired, that is, one which can be applied to any of several systems of a given class.

6. The inner workings of the system are inaccessible, and hence, a plausible model can only be inferred from the observed behavior of the system.

As a qualitative model, the fuzzy cognitive map is capable of predicting future states of a system. With this capability, there are two primary application areas for fuzzy cognitive maps:

1. In the monitoring of complex systems, where the fuzzy cognitive map is continually predicting future system states.

2. In the planning operations of a mission, where the fuzzy cognitive map is used to predict the likely outcome of a proposed action.

For the system monitoring application, the current states of the system(s) that are being monitored are periodically loaded into the fuzzy cognitive map, and the fuzzy cognitive map is then allowed to evolve forward in time at a rapid rate (faster than real time). In this way, the likely future states of the system can be predicted. The predicted states may be used, for example, to automatically trigger some alarm signals and corrective action may be taken before a truly critical system state arises.

Furthermore, for this real time monitoring and predicting application, the fuzzy cognitive map has the potential to significantly increase accuracy and reduce human workloads. The fuzzy cognitive map is able to follow large numbers of interactions in complex situations without being overwhelmed by too much data. It, therefore, has the potential to greatly reduce the amount of data that must be considered by the human operators of the system. Unlike human beings, the fuzzy cognitive map is not affected by emotional stress in crisis situations, nor is it subject to distraction or fatigue.

For the planning and prediction applications, hypothetical system states are loaded into the fuzzy cognitive map, and the fuzzy cognitive map is then allowed to evolve forward in time. The predicted future states may be examined to determine the likely outcome of placing the system in the hypothetical state. For this kind of applications, the fuzzy cognitive map can serve as a planning aid.

Fuzzy cognitive maps can serve as aids in both near-term and long-term planning. An example of near-term planning is using the fuzzy cognitive map to aid in finding a corrective action in the event of a system malfunction. Upon system malfunction, the current state of the system is loaded into the fuzzy cognitive map and a plausible corrective action in the form of a forcing input is applied to the fuzzy cognitive map. The fuzzy cognitive map is then allowed to evolve forward in time to predict the likely outcome of the corrective action. If the predicted outcome is acceptable, the corrective action may then be applied to the real system. The process may be repeated should the evolution of the real system deviate from the previous prediction. This fuzzy cognitive map application example resembles the likely mental process of a human operator of a complex system. That is, before the operator applies the corrective action, he or she will try to predict the outcome by mentally modeling the evolution of the system. Several plausible corrective actions may be considered by the operator before one leading to an acceptable predicted outcome is found.

Furthermore, a fuzzy cognitive map may be used, for example, in the design phase to predict the consequences of component or subsystem failure, thereby leading to safer system designs. Another use is as a system simulator for training purposes. It is conceivable too, that the fuzzy cognitive map might find use in long-term strategic planning and policy making.

The fuzzy cognitive map offers a very important application related property which is not fully addressed by other knowledge representation methods: it can be utilized to perform qualitative causal modeling, that is, emulate behavior of a system through qualitative causal modeling. The system may be one for which it is infeasible to produce a full quantitative model because: 1) the system is too complex to be modeled in detail in a reasonable amount of time, 2) the system is subject to unpredictable variation which would render any quantitative model inexact, or 3) the inner workings of the system are inaccessible, and hence, a plausible model can only be inferred from the observed behavior of the system.

As a qualitative causal model, according to the knowledge embedded in the map, the fuzzy cognitive map is capable of doing the following functions:

1. Classify or discriminate the essential variables among a large set of data variables, and thus correct decisions can be made on the actions that should be taken by the system or a human operator.

2. Predict the future status of a system. With this capability, fuzzy cognitive maps can be used to predict the likely outcome of a proposed action.

3. Identify the most likely causes for a specific system status. This is especially important for a system whose overall status is a result of propagating and magnifying of some minor problems which occurred on a small fraction of the system.

Based on the discussions shown above, it is obvious that a decision making system based on fuzzy cognitive maps will fulfill the characteristic requirements of a reliable and robust decision making system.

We next present the fuzzy logic inferencing (modus ponens) of Lukasiewicz and then compare it to the fuzzy inferencing rule of Gaines. We find that either form of inferencing may be used in the fuzzy cognitive map, though the latter has a more natural (and conventional) implementation.

Lukasiewicz Inferencing Rule

The Polish logician Lukasiewicz developed a logical system in which truth values are extended continuously over the range [0, 1]. In this system the implication operator is defined as

$$t_L\,(A \to B) \;=\; [1, 1 - t(A) + t(B)] \tag{1}$$

where t(A) and t(B) are the truth values of statements A and B, respectively. To derive the fuzzy inferencing rule corresponding to classical modus ponens we take

$$t_L\,(A \to B) \;=\; c,\; 0 \le c \le 1 \tag{2}$$

and

$$t(A) \ge a,\; 0 \le a \le 1 \tag{3}$$

and then solve for truth value t(B):

Case 1: $c < 1$

$$\begin{aligned} 1 - t(A) + t(B) &= c,\ then \\ t(B) &= t(A) + c - 1 \ge a + c + 1 \end{aligned} \tag{4}$$

Case 2: $c = 1$

$$
\begin{aligned}
&1 - t(A) = t(B) \geq 1, \; then \\
&t(B) \geq t(A) \; provided \; t(B) \geq 0, \; then \\
&t(B) \geq a \; = \; a + c - 1
\end{aligned}
\tag{5}
$$

Therefore, the truth value t(B) may be expressed by

$$t(B) \geq \; \max \; (0, \mathrm{a} + \mathrm{c} - 1) \tag{6}$$

This result reduces to classical modus ponens for bivalent truth values a and c.

Gaines Inferencing Rule

The Gaines implication operator is defined as

$$t_G\,(A \rightarrow B) = \begin{Bmatrix} \min \; (1, t(B) / t(A)) & if\, t\,(A) > 0 \\ 1 & if\, t\,(\mathrm{A}) + 0 \end{Bmatrix} \tag{7}$$

as before we take and then solve for the truth value t(B):

Case 1:t(A) > 0

$$
\begin{aligned}
&t(B) / t(A) = c, \; then \\
&t(B) = t(A) \cdot c \geq a \cdot c
\end{aligned}
\tag{8}
$$

Case 2:t(A) = 0

$$
\begin{aligned}
&c = 1 \; and \; a = 0 \\
&t(B) \geq 0 = a \cdot c
\end{aligned}
\tag{9}
$$

therefore the final result is

$$t(B) \geq a \cdot c \tag{10}$$

The last result gives theoretical justification for the use of multiplication as the fundamental inferencing operator in the fuzzy cognitive map.

Inference Example

For the purpose of demonstrating the fuzzy cognitive map inference process, the example shown in [20] is utilized here. Figure 7 displays a fuzzy cognitive map with causal links between ice, ozone, and chlorine monoxide in the atmosphere, and Figure 8 shows the matrix representation of the fuzzy cognitive map.

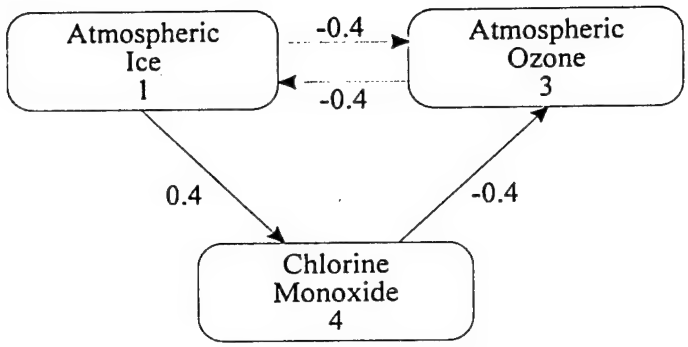


Figure 7. Fuzzy cognitive map for ice, ozone, and chlorine monoxide in the atmosphere.

Atmospheric ice tends to decrease ozone and to increase the level of chlorine monoxide. Chlorine monoxide tends to decrease the amount of ozone. An increase in ozone decreases the amount of atmospheric ice. The numbers near the links are causal strengths. The sign indicates whether the cause (origin of an arrow) increases (+) or decreases (-) the effect (end of arrow). The magnitude ranges from 0 to 1 with 0 meaning no causal linkage and 1 meaning the effect is strong. The variables (or concepts) of the fuzzy cognitive map are discrete with values -1 (low), 0 (normal), or 1 (high). For example, the state vector (-1, 0, +1) means ice is low, chlorine monoxide is normal, and ozone is high.

$$M = \begin{bmatrix} 0.0 & 0.4 & -0.4 \\ 0.0 & 0.0 & -0.4 \\ -0.4 & 0.0 & 0.0 \end{bmatrix}$$

Figure 8. Matrix representation of the fuzzy cognitive map shown in Figure 7.

The fuzzy cognitive map inference procedure has five steps and results in stable states or dynamic equilibrium depending upon the matrix and the initial conditions, and the inference procedure recalls these hidden patterns given an initial states. The five steps of the fuzzy cognitive map inference procedure are:

1. Form the current state vector.

2. Multiply the state vector and the fuzzy cognitive map matrix giving new temporary variable values.

3. Apply the threshold functions to every temporary variable in the vector:

   a. if variable <= -0.33 then variable = -1.

   b. if variable > -0.33 and < +0.33 then variable = 0.

   c. if variable >= +0.33 then variable = +1.

4. Compose the new state vector.

5. Return to step 2 until the state stabilizes to a constant vector or a state repeats.

Suppose the initial state of the atmosphere is the vector (0, +1, 0), that is, normal ice, high chlorine monoxide, and normal ozone. The states listed in Table 1 are the results of inference processing for the fuzzy cognitive map shown in Figure 7.

The net effect (+1, +1, -1) means the atmosphere system has more than normal amounts of ice, more than normal chlorine monoxide, and low ozone. According to this abbreviated model, ozone depletion arises from a high level of chlorine monoxide. The state vectors:

(0, +1, 0), (0, 0, -1), (+1, 0, 0), (0, +1, -1), and (1, 0, -1)

are the transition states. They lead to a stable state -- a fixed point limit cycle (+1, +1, -1).

Table 1: The effects of high chlorine monoxide in the atmosphere.

Iteration	Input Vector	Output Vector
1	(0, +1, 0)	(0, 0, -1)
2	(0, 0, -1)	(+1, 0, 0)
3	(+1, 0, 0)	(0, +1, -1)
4	(0, +1, -1)	(+1, 0, -1)
5	(+1, 0, -1)	(+1, +1, -1)
6	(+1, +1, -1)	(+1, +1, -1)

FCM Time Evolution

To define the time evolution of the FCM, we choose the dynamical equation

$$c_j \cdot \dot{a}_j = \sum_i S(a_i)W_{ij} - \frac{a_j}{R_j} + I_j \tag{11}$$

where a_j is the activation of the jth node, S is a signal function with a range in [0,1] (usually a sigmoid), w_{ij} is the strength of the connection from node i to node j (may be positive or negative), R_j is a positive constant which controls the decay rate of the jth node's activation, I_j is an external input to the jth node, and c_j is a positive constant controlling the slew rate of the jth nodes' activation. With this formulation the dynamical behavior of the FCM becomes that of the continuous Hopfield network. A more general form of the dynamics may be given by

$$\dot{a}_j = -\alpha(a_j)[\beta(a_j) - \sum_i S(a_j)W_{ij}] \tag{12}$$

where α is a nonnegative amplification function and β is an arbitrary continuous function. The later form was introduced and studied by Cohen and Grossberg. Although the more general form may find use in FCM's, the present work will focus on equation (1).

FCM Training

In this section we consider FCM training - the adjustment of the causal interconnections, either off-line or in real time, so as to improve performance. A principal finding, to be presented below, in that it is important to distinguish between two types of training: refinement of existing FCM's and the growing of entirely new FCM's. A learning law appropriate for one of these might not be suitable for the other. In general, the growing of new FCM's will resemble the training of neural network and will require a large set of training examples. The refinement of existing FCM's, on the other hand may include those cases where a small amount of new data is used to refine an FCM which is already in use.

Hebbian Learning

The Hebbian learning law, in its simplest mathematical form, may be expressed as

$$\Delta W_{ij} = C_i C_j \tag{13}$$

It is also often expressed in the form

$$\dot{W}_{ij} = -W_{ij} + C_i C_j \tag{14}$$

where the first term on the right gives a decay or "forgetting." Implicit in equation (14) is a coefficient to set the decay rate.

Hebbian learning in the form presented above is inadequate for FCM training since there is an incompatibility between equation (13) and the basic FCM structure which has bipolar interconnections and unipolar nodes. Without modification, equation (13) can lead only to increases in the interconnection strengths.

A straightforward way to avoid the above mentioned incompatibility is to employ the differential Hebbian learning law:

$$\dot{W}_{ij} = -W_{ij} + \dot{C}_i \dot{C}_j \tag{15}$$

The differential Hebbian learning law is based on concomitant variation; if C_i increases concomitantly with C_j then there is likely to be a causal connection between C_i and C_j.

We note here that the Hebbian laws are symmetric. That is, growing a causal connection from C_i to C_j using eq. (14) or eq. (15) will also grow a causal link from C_j to C_i. This is undesirable. Fire is likely to cause smoke but smoke is not likely to cause fire. Fortunately, this situation can be mitigated by using differential Hebbian learning with lagged variation:

$$\Delta W_{ij}(t) = \Delta C_i(t-1) \cdot \Delta C_j(t)$$

For training or refining the FCM Hebbian learning has certain disadvantages. This becomes evident when one uses Hebbian learning to refine an existing FCM, such as one which has been abstracted from first principles. Such an FCM usually has relatively few causal interconnects (most of the elements of the interconnection matrix are zero). Under Hebbian learning causal interconnects proliferate rapidly. Even after a short period of adaption most of the interconnection matrix elements become non-zero. In principle, over a sufficiently long training period, those edges for which there is no causal connection should average to zero. However, long training periods are not desirable for FCM refinement. In fact, if the training period is sufficiently long to drive the non-causal edges to zero then it is probably also long enough to grow the FCM from scratch. Clearly, a different learning rule is desired for the refinement of existing FCMs. One candidate is Klopf's drive-reinforcement model.

Klopf's drive-reinforcement

A simplified form of the Klopf drive-reinforcement model can be expressed as

$$\dot{W}_{ij} = -W_{ij} + |W_{ij}| C_i\, C_j \tag{16}$$

or in lagged form (without decay) as

$$\Delta W_{ij}\ (t) = \left| W_{ij}\ (t-1) \right| \Delta C_i\ (t-1) \cdot \Delta C_j\ (t) \tag{17}$$

The chief advantage of this rule in an FCM context is that only non-zero edges are altered. It therefore is well suited to the adaptive refinement of already existing FCMs, though it may not be appropriate for growing FCM's from scratch.

We conclude this section by summarizing the main findings about FCM learning:

1. Simple Hebbian learning is not suitable for training FCMs having the structure presented here since it is incompatible with the use of bipolar interconnects and unipolar nodes.

2. Differential Hebbian learning is compatible with the FCM structure and is appropriate for growing new FCM's from scratch. If used to refine existing FCMs, this learning rule may generate spurious causal connections.

3. A version of Klopf's drive reinforcement is probably the best learning rule for the refinement of existing FCMs, since it alters only those causal connections which are already non-zero and therefore will not lead to a proliferation of possibly spurious connections.

For Marine Corp application, the FCM approach needs to be developed in a software module to demonstrate a command and control application.

Task 2. Develop a Prototype Knowledge-based System Module in Software

With the Microsoft® Visual C++™ language, a stand-alone software module, called TACAN's FCM ToolKit™ Software, used for the formation, configuration and manipulation of fuzzy cognitive maps was designed and created.

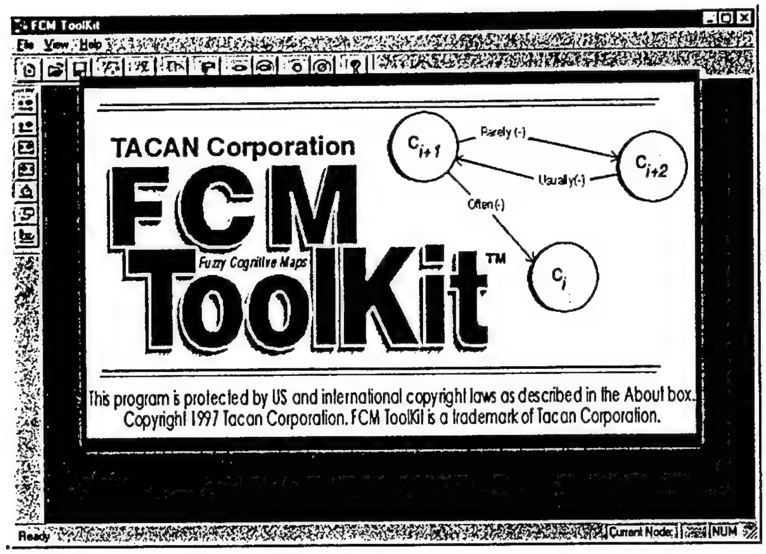


Figure 9. C++ FCM ToolKit™.

The FCM ToolKit™ Software has the following functions/features:

1. Allows user to create and manipulate fuzzy cognitive maps.
2. Synthesizes different fuzzy cognitive maps into a new fuzzy cognitive map.
3. Defines hierarchical fuzzy cognitive maps structure.
4. A friendly Windows™ graphical user interface.
5. Users can drag-and-drop the nodes in a fuzzy cognitive map freely to design the topology of the map in the way they like.
6. The nodes and links in a fuzzy cognitive map can be freely modified, added or deleted.
7. With a defined fuzzy cognitive map and the initial states of the map nodes, users can run continuous causal inference simulations on the currently opened fuzzy cognitive map.

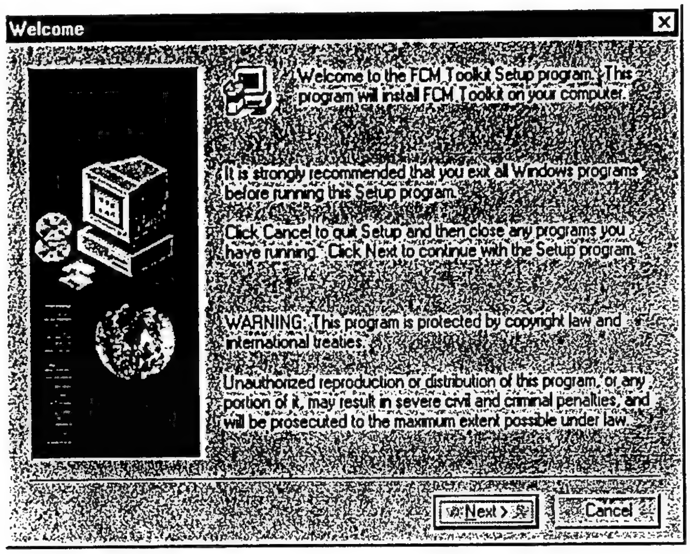
Install the FCM ToolKit™ Software

The FCM ToolKit™ Software is written with the Microsoft® Visual C++™ language, and it can be run on a computer system with the Microsoft® Windows95™ operating system.

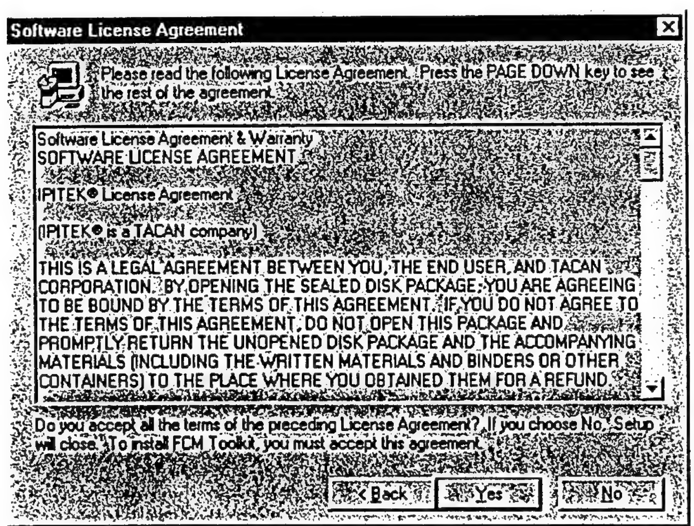
Getting Started With the FCM ToolKit™ Software

Double click on the program icon that is created in "Install" to start the FCM ToolKit™ Software. Then the main window of the FCM ToolKit™ Software appears, shown in Figure 10.

Figure 11 and Figure 12 show additional screens that are used to start the FCM ToolKitTM during the installation of the software.

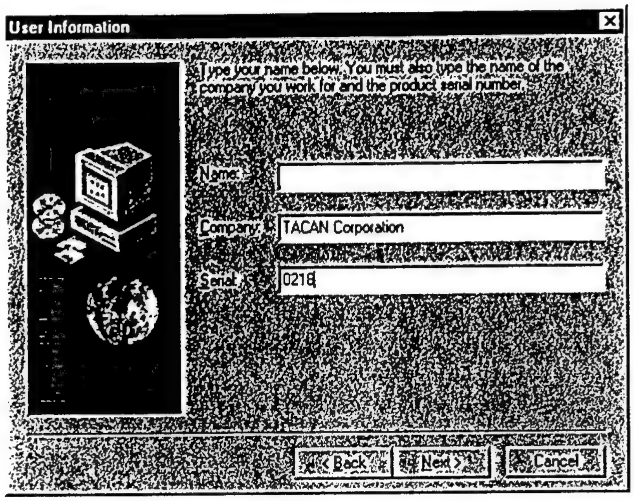


10. (a)

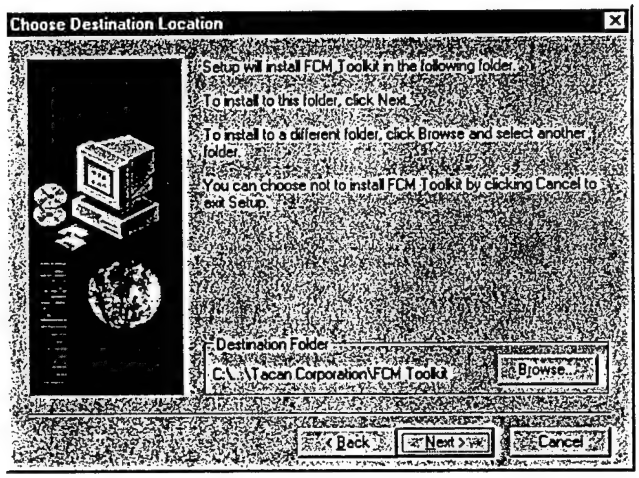


10. (b)

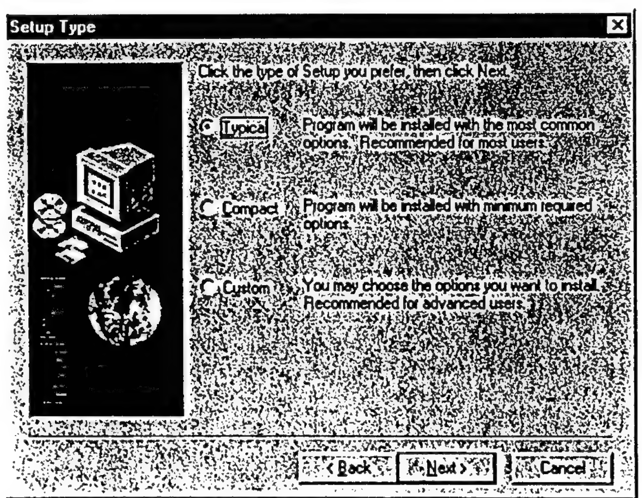
Figure 10. (a) FCM ToolKit^{TM} Main Window, and (b) License Agreement During Installation.



11. (a)



11. (b)



11. (c)

Figure 11. (a) User Information Screen, (b) File Location Screen, and (c) Setup Screen During Installation.

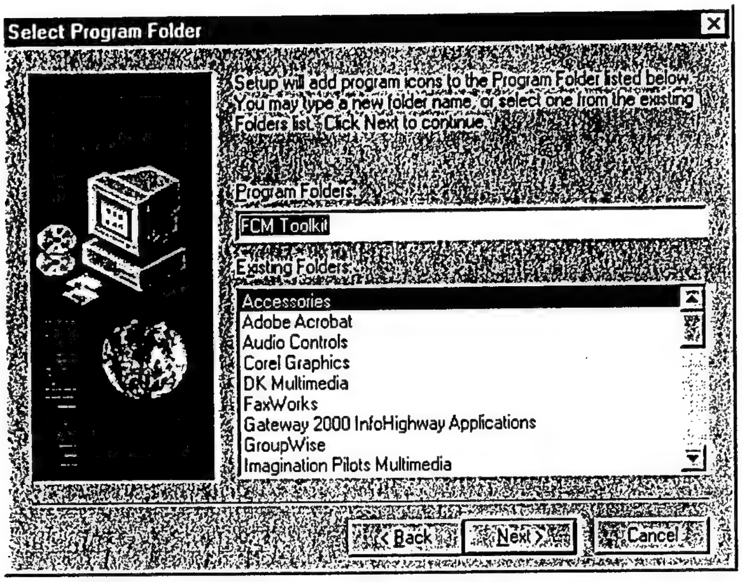


Figure 12. Select Program Folder Menu.

(Note: if you have the FCM ToolKit™ Software with Target Simulation Module then the "MarineCorpsSimulation.fcm" file must be loaded prior to using the Simulation Module.)

CREATING A NEW FCM FILE

The following sections show the creation of a simple three node FCM:

The "File, New" Command

Figure 13. Starting a New File.

Select this command to create a new fuzzy cognitive map, see Figure 13. After this command is selected, a series of dialog boxes (Figures 14-17) will be presented to the user to configure a new FCM with 2 nodes interconnected with a link weight.

Initially, the user is prompted for the parameters of the "source" node. (This dialog can be accessed later by double-clicking within the graphical image (depicted by a colored circle) of any node.)

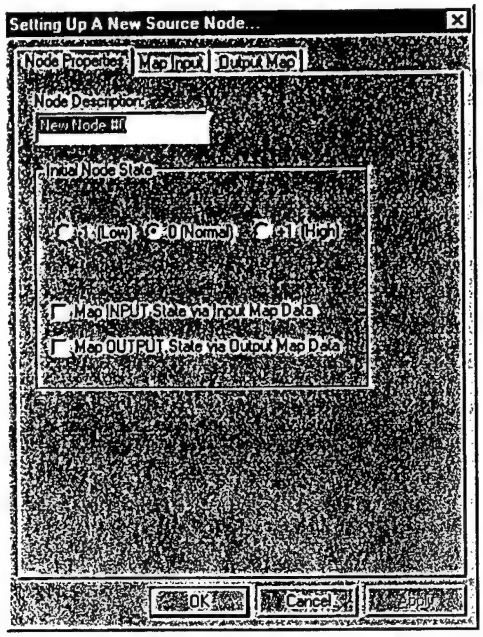


Figure 14. Node Description and Node State.

The node properties tab allows the user to specify the node description, initial node state and I/O mapping parameters.

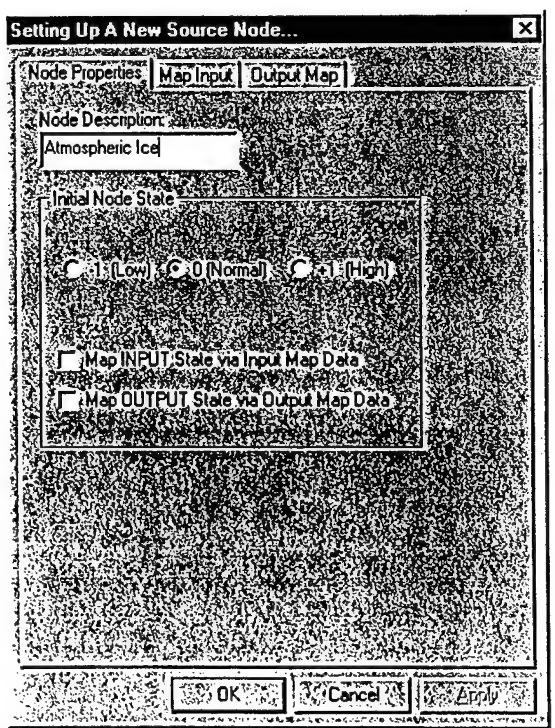


Figure 15. Application of Source Node Setup.

After setting up the source node, the user is then prompted to enter the "Destination" node properties.

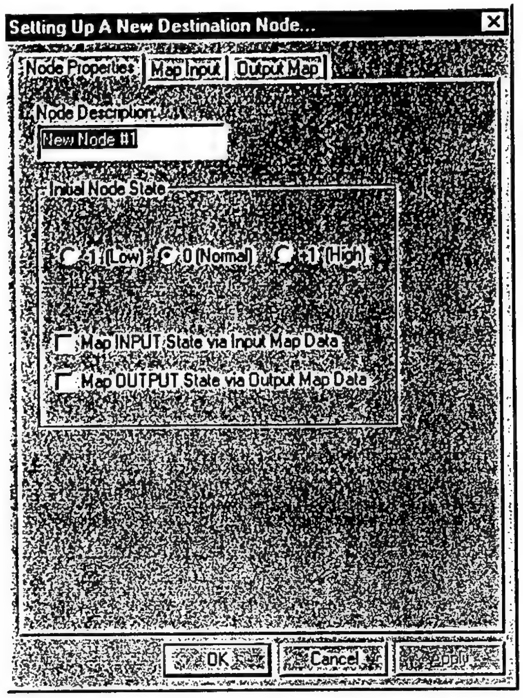


Figure 16. Destination Node Setup.

Once the initial node properties have been configured for the first two nodes, the user is prompted to enter the link weight.

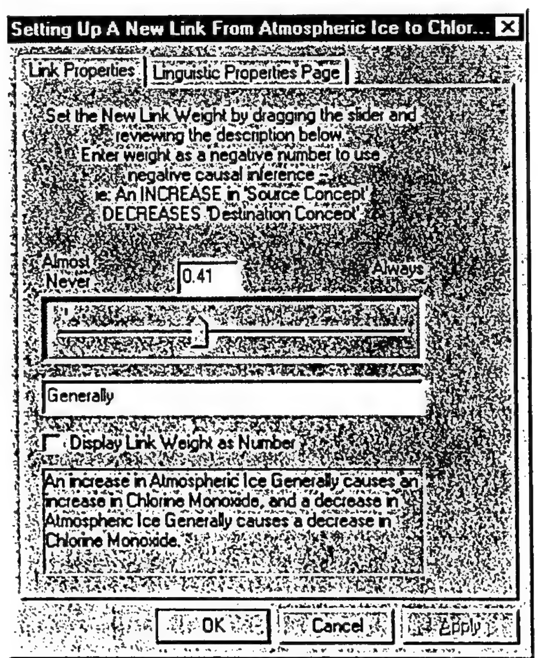


Figure 17. Link Weight Setup.

In the Link Properties tab, the user has the ability to modify the link weight value and display format of the link weight between 2 nodes. (This dialog can be accessed later by double-clicking on the link weight text.)

Users can either enter a numeric link weight in the range of 0.0 to 1.0 directly into the text box, or use the slider bar to adjust the weight. If you enter a value outside of this range you will see the following message (Figure 18):

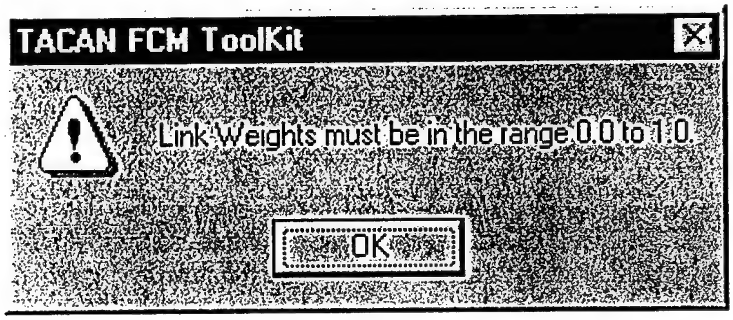


Figure 18. Link Weight Reminder.

Note that by making this weight negative, you tell the FCM Inferencing engine to us negative causal inference between these two nodes.

A brief description of the causal effect that the source node has on the destination node is displayed at the bottom of the dialog box.

By clicking the "Display Link Weight as Number" check box, you modify the way the link weight is displayed within the FCM window. Checking the box causes the numeric value to be displayed, while not checking the box tells the system to use the linguistic term in the graphical display of the link.

After clicking OK, the ToolKit displays the graphical representation of the FCM (topology map).

Inserting a New Node into an Existing FCM

There are a few methods the user may use to insert a new node into an existing FCM:

1) Select "Insert" from the "Nodes" menu.

2) Depress the Insert Node button on the tool bar (Figure 19).

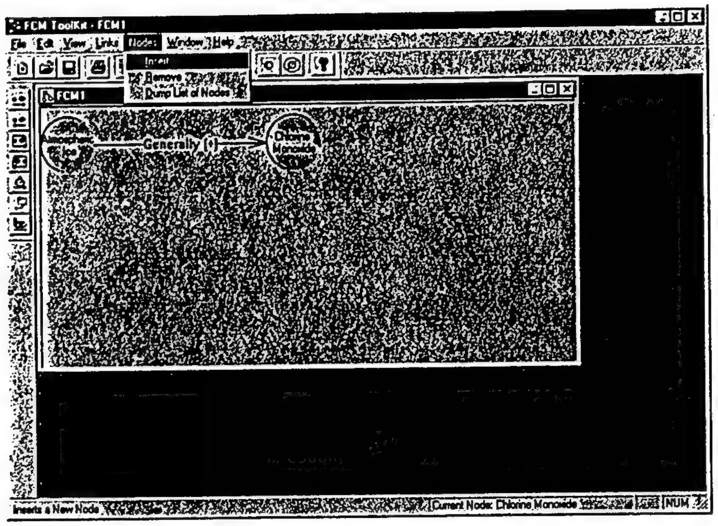


Figure 19. Inserting a New Node.

After executing either of the above commands the FCM ToolKit will display the Node Properties dialog box to allow configuration of the new node. When done, the user clicks the OK button and is able to place the new node on the map (Figure 20).

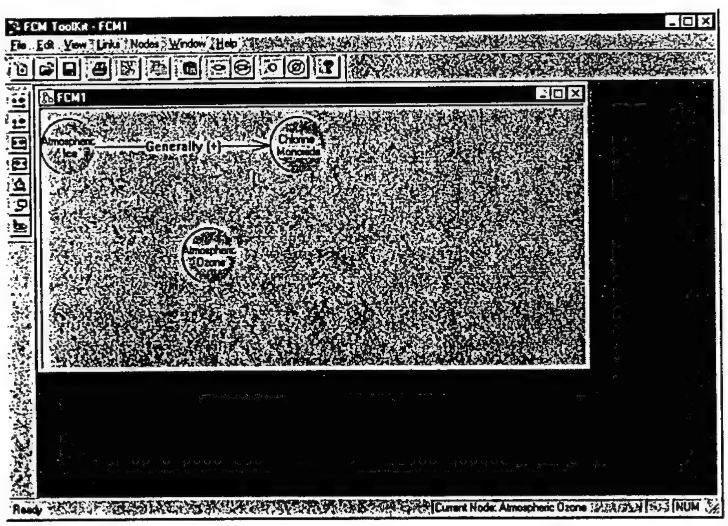


Figure 20. New Node Inserted.

Inserting a Link Between Two Nodes

There are a few methods the user may use to insert a link weight between two nodes:

1) Select "Insert" from the "Links" menu.

2) Depress the Insert Link button on the tool bar (Figure 21).

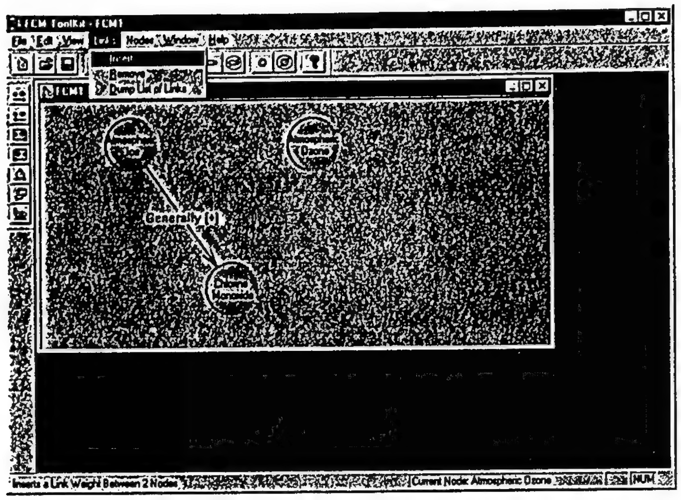


Figure 21. Inserting a Link.

After executing either of the above commands, the FCM ToolKit will display a confirming dialog box (Figure 22) that gives the user a chance to change his mind about connecting two nodes together.

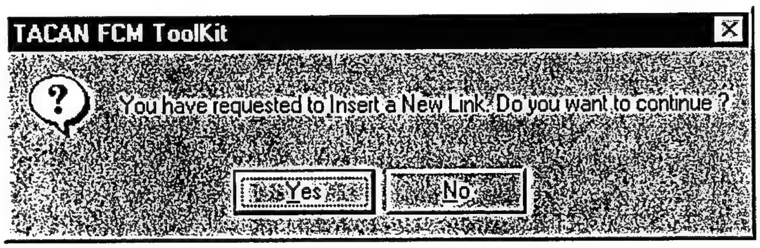


Figure 22. Confirmation Box for New Link.

Upon confirmation, the FCM ToolKit then prompts the user to indicate the source node (Figure 23) by clicking on the appropriate node.

Figure 23. Prompt for Source Node.

Next, the user is prompted to click on the destination node (Figure 24).

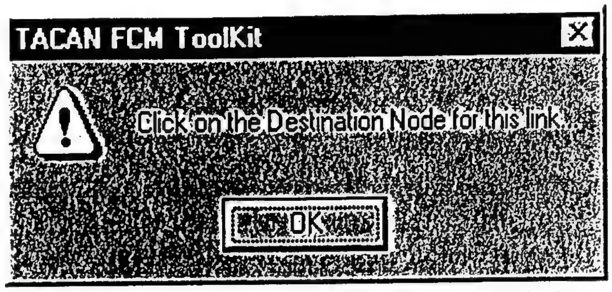


Figure 24. Prompt for Destination Node.

Upon clicking on the destination node, the FCM ToolKit will display the Link properties (Figure 25) dialog box.

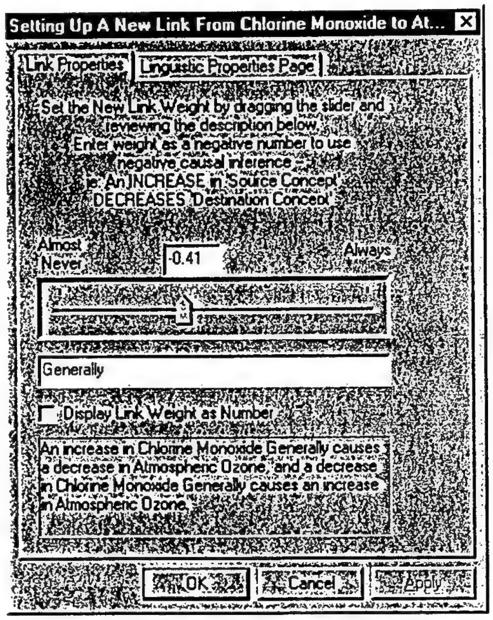


Figure 25. Link Properties Screen.

After clicking OK, the FCM ToolKit again shows the topology map (Figure 26) with the newly inserted link weight.

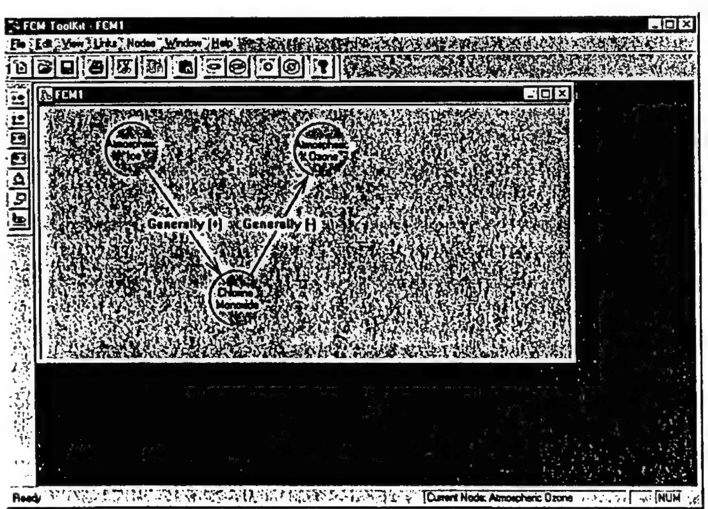


Figure 26. FCM Graph.

Creating Bidirectional Links

The sample FCM requires a bidirectional link between two of the nodes. To create a bidirectional link, use one of the methods for inserting a link as described above, with one of the nodes as the source node, the other as the destination node. Then repeat the insert link process again, only this time use the previously selected destination node as the source. (i.e., if node A was the source and node B was the destination for the first link, then make node B the source and node A the destination for the second link.) Note you may need to reposition the nodes on the screen to make bidirectional links easier to read (Figure 27).

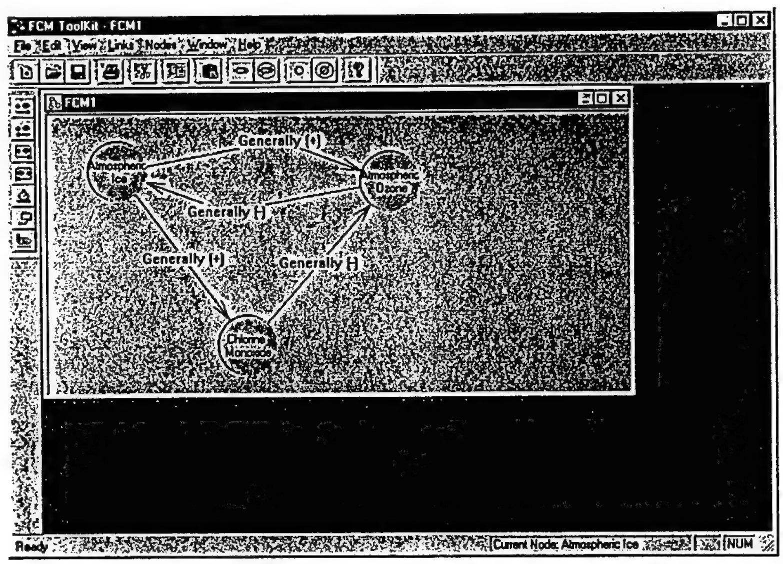


Figure 27. FCM Graph With Bidirectional Links.

Alternative Node/Link Insert Method

There is an alternative shortcut method for simultaneously inserting a new node and creating a link to this node:

While holding down the Ctrl. key, click on a node you consider to be the source node, and start dragging the node. This will "peel off" a new node from the source node. A new node will be placed on the map when you release the mouse button, and the Node Properties dialog box will be displayed allowing the user to configure the new node. Once configured, the user clicks OK and the Link properties dialog box is automatically displayed allowing the link weight between the original (source) node and the new(destination) node to be set.

Saving the newly created FCM

To save the FCM, select Save As from the File menu to display the Save As dialog box (Figure 28).

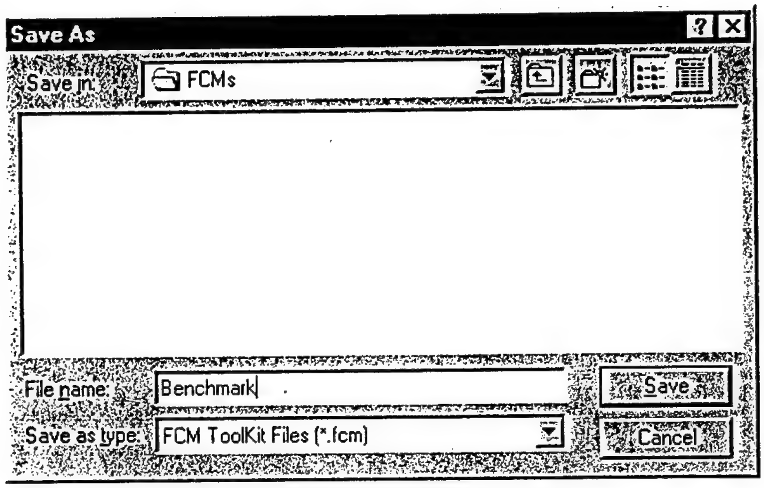


Figure 28. Save FCM Screen.

This is the standard Windows95 Save As dialog box, which allows the user to navigate to an existing folder or create a new folder, and specify the name of the FCM file.

FCM Inference Simulations

One of the most powerful advantages of the TACAN's Visual C++ FCM ToolKit is that it allows user step forward (prediction) and backward(diagnostics) through node states until an equilibrium or limit cycle has been reached (Figure 29).

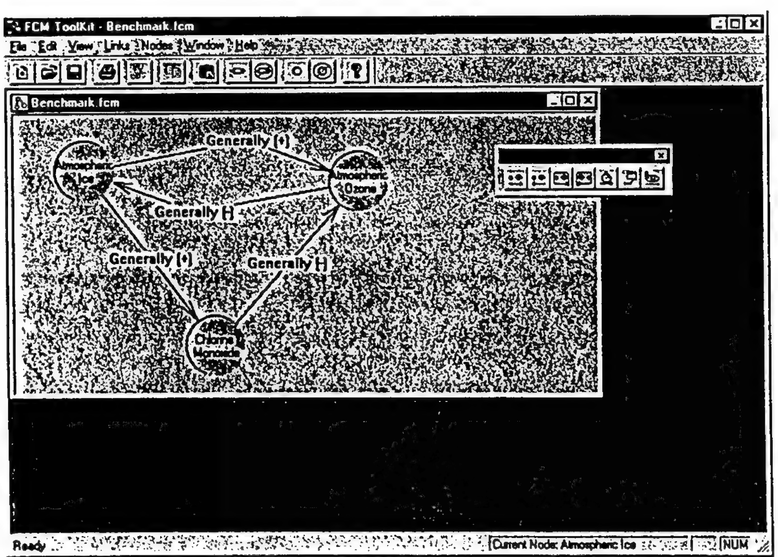


Figure 29. FCM With Simulation.

In the current version of the ToolKit, inferred states are output to a text file which contains the following information:

1) The initial input vector

2) The FCM matrix used during the inferencing process

3) The series of inferred output vectors

To run a forward inference with the ToolKit, first create a FCM and save it. Figure 30 below shows the Simulation Toolbar.

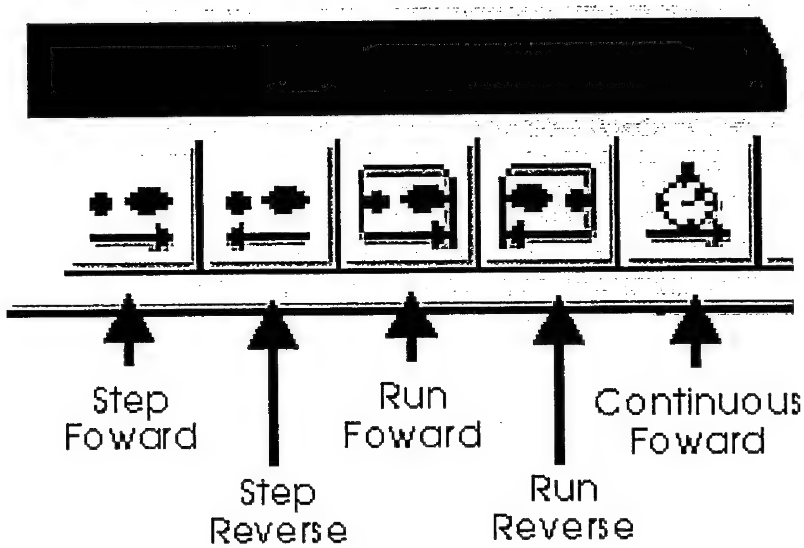


Figure 30. Simulation Toolbar.

The first two buttons allow the user to single step forward or reverse to test what the next or previous state will be, based upon the current state.

The next two buttons allow the FCM to run until a limit cycle is detected, showing all state transitions arising from the initial state, or in the case of reverse inferencing, showing all states that could lead to the initial state.

Figure 31, below, shows the text output from running the Benchmark.fcm example in the forward inference mode.

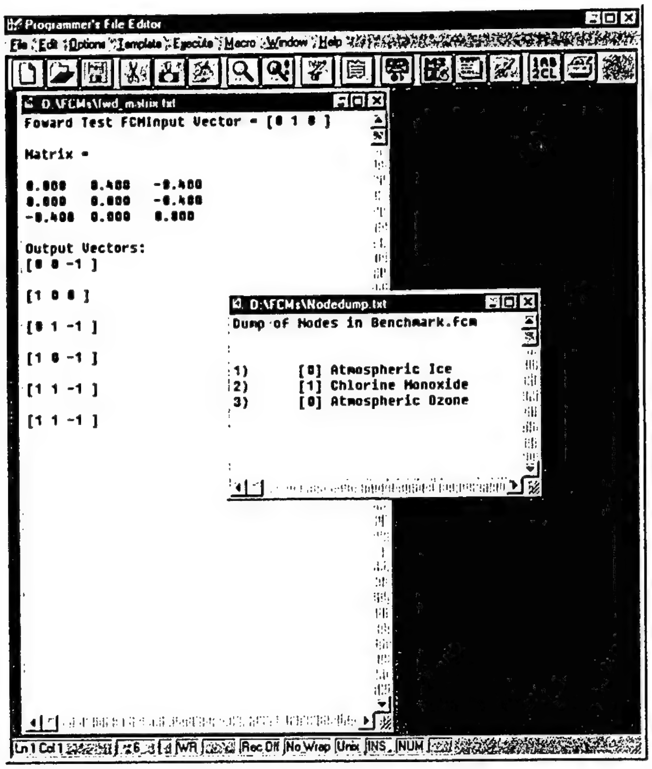


Figure 31. Example of Forward Inferencing.

The initial input vector of [0 1 0] indicates we are starting with normal ice, high chlorine monoxide, and normal ozone. After running the forward inference command, we see the net effect or equilibrium state of [1 1 -1], indicating the atmosphere system has more than normal amounts of ice and chlorine monoxide, and low ozone.

The state vectors

[0 0 -1],

[1 0 0],

[0 1 -1], and

[1 0 -1]

are transition states. They lead to the stable, fixed point limit cycle of [1 1 -1].

Figure 32 shows the text output from running the Benchmark.fcm example in the reverse inference mode.

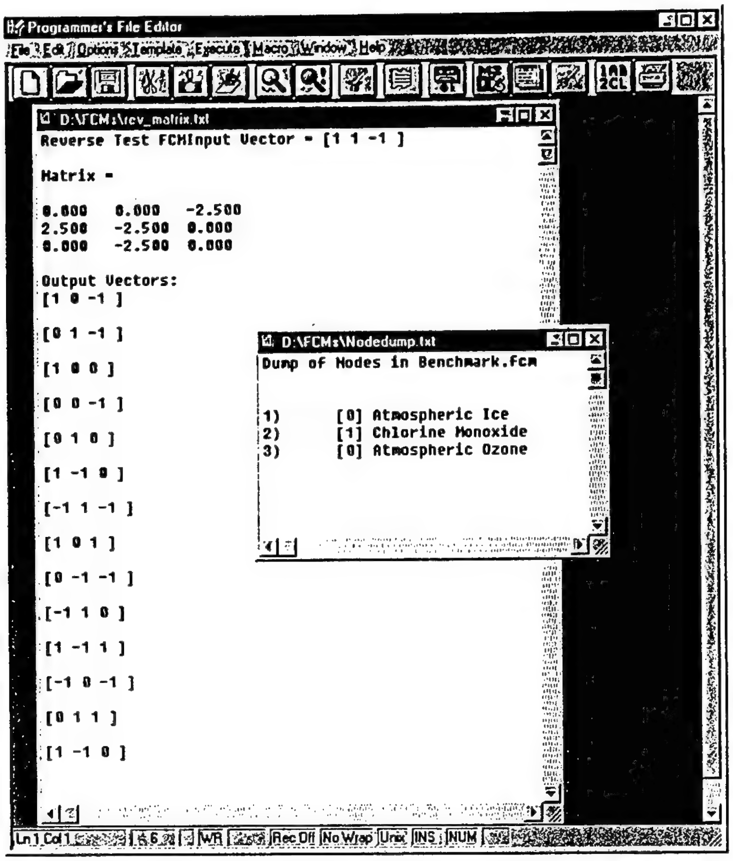


Figure 32. Example of Reverse Inferencing.

With the inverse matrix, the possible causes of the stable states in the atmosphere [1 1 -1], i.e., high ice and chlorine monoxide and low ozone in the atmosphere may be identified.

The initial input vector of [1 1 -1] indicates we are starting with high ice, high chlorine monoxide, and low ozone. Apparently, any of the 13 vectors shown represent the states of ice, chlorine monoxide and ozone in the atmosphere that will eventually deplete the ozone content in the atmosphere.

Task 3. Design a concept for command and control with Marine Corps applications

Marine Corps Simulation Module

A simple FCM was created to demonstrate an application of FCM algorithms to command and control as illustrated in Figure 33.

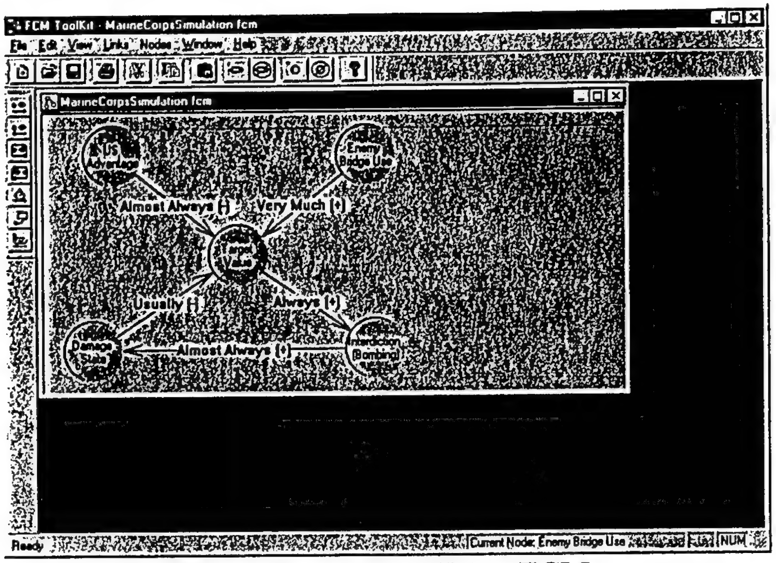


Figure 33. Target Value FCM.

This FCM uses five nodes to very basically describe the information needed to determine whether or not a particular bridge is to be considered a valid target based upon the current state of the animation.

An animated simulation engine was written using the Microsoft® VisualBasic™ language to facilitate rapid development. This animation communicates with the Visual C++ FCM ToolKit using the standard Windows DDEML Dynamic Data Exchange Management Library. The scenario behind the simulation is that a series of bridges is approached by enemy troops. If there is no U.S. Advantage to NOT destroying the bridge, and if the bridge is not currently damaged, then the bridge is a valid target and should be destroyed.

The enemy troop position, bridge locations and U.S. Advantage can all be changed by the user while the simulation is running, by dragging and dropping as illustrated in Figure 34. When running, the simulator selects a target to test, indicated by a rectangle drawn around the bridge (see Figure 35), and forms the input vector for the Enemy Bridge Use, U.S. Advantage, and Damage State nodes of the FCM.

The relative distance of the Enemy Troops from the bridge being tested is mapped into the range of [-1 0 1] where 1 indicates the troops are possibly using the bridge. The U.S. Advantage state is formed in the same way, where a 1 state indicates some reason NOT to destroy the given bridge.

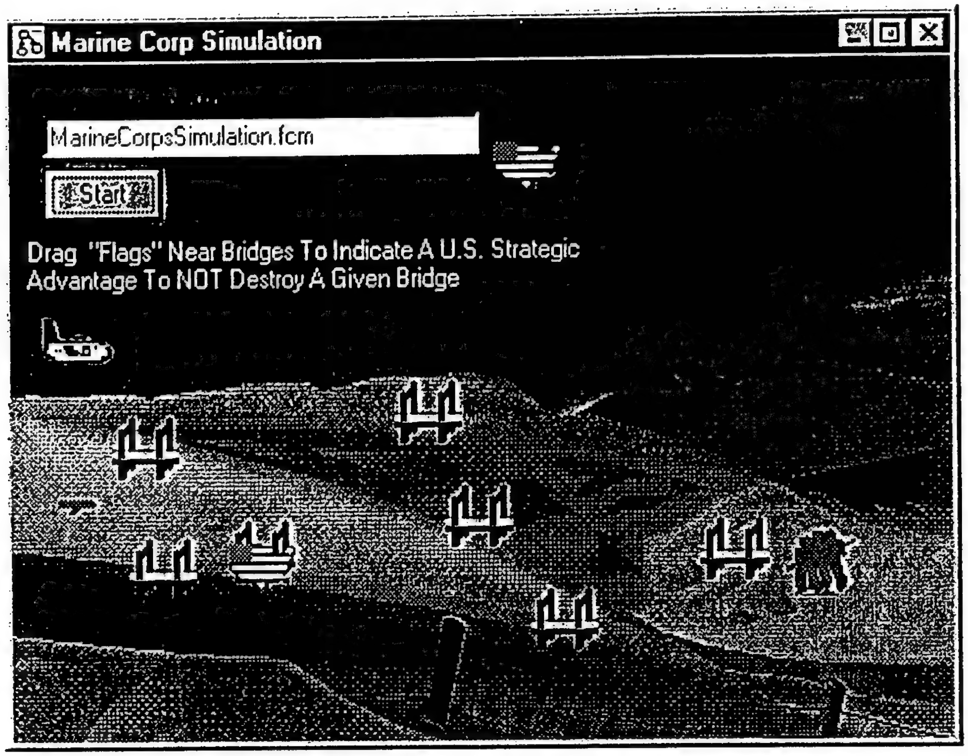


Figure 34. Command and Control Simulation with FCM.

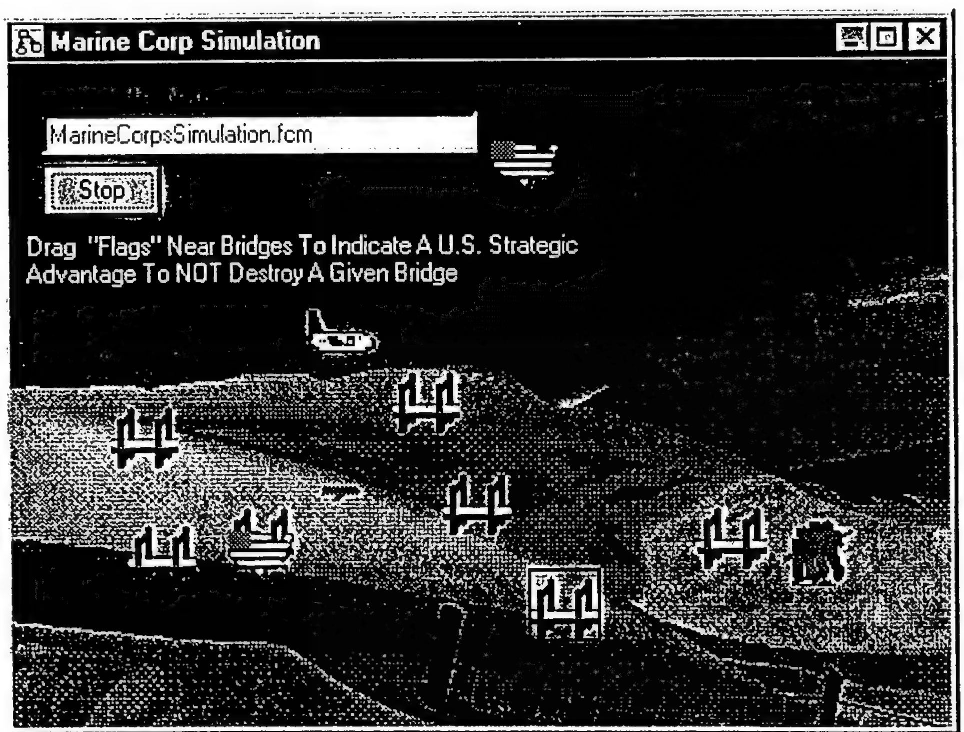


Figure 35. Simulation Selects a Specific Target, Shown by Rectangle.

A function call is then made to the FCM ToolKit, which tells it to perform the Step Forward Inference algorithm. As the algorithm executes, the ToolKit queries the simulator for the initial

node states. The inferencing engine is then iterated just once to find what the next state will be based upon the current input vectors and the configuration of the FCM.

The FCM ToolKit then returns the vector of the Target Value node to the simulator. If there is no target value to this bridge, (vector of 0 or -1), the simulation advances to the next bridge and the process is started again. Once all bridges have been tested, the simulator loops back to the first bridge and continues iterating through all bridges until the simulation is topped.

If the conditions are such that the ToolKit returns a Valid Target state (1) to the simulator (Figure 36), the simulation sends a bomb (Figure 37) to destroy the bridge (Figure 38) and Damage State for that bridge is set to High.

According to the FCM, the next time this bridge is tested, it should not be bombed because the Damage State node for that bridge is now High(1). Note: the user can double click on any bridge at any time to toggle its state from damaged to good, and vice versa, to see the effect. (Make a bridge good, then it's a candidate for bombing, make it damaged and no bombing should occur.) Users can also drag and drop the flags on a bridge to indicate that there is some Advantage to NOT destroy a given bridge (political climate, strategic advantage, etc.), see Figure 39.

Figure 40 shows a new target acquired with a large value and thus being bombed. [INSERT FCM26.bmp]

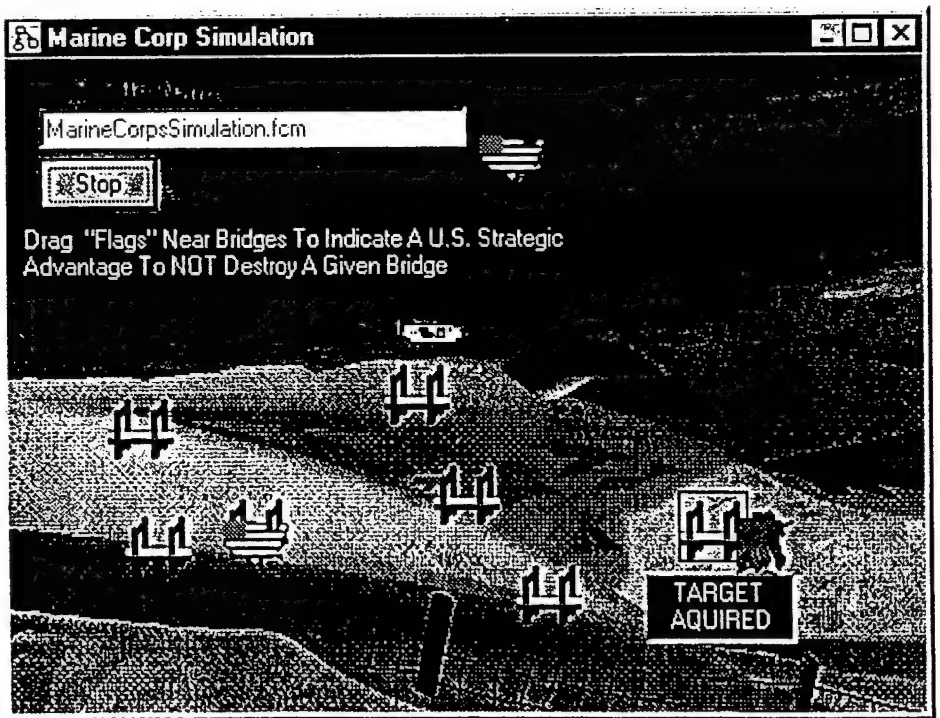


Figure 36. Simulation Shows Target With Large Value.

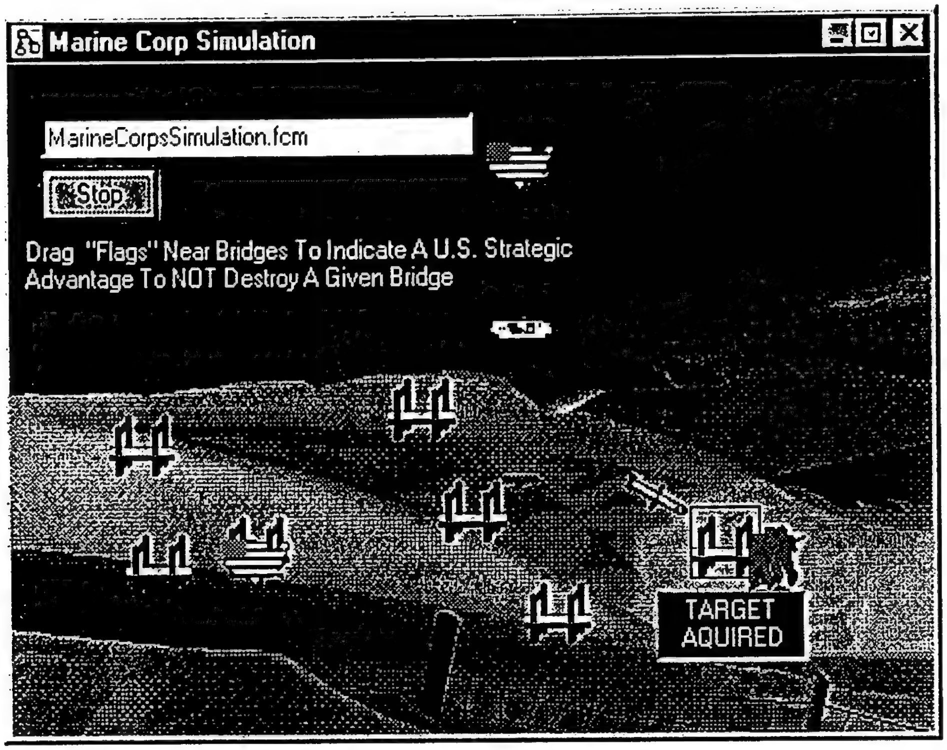


Figure 37. Simulation Shows Target Being Bombed.

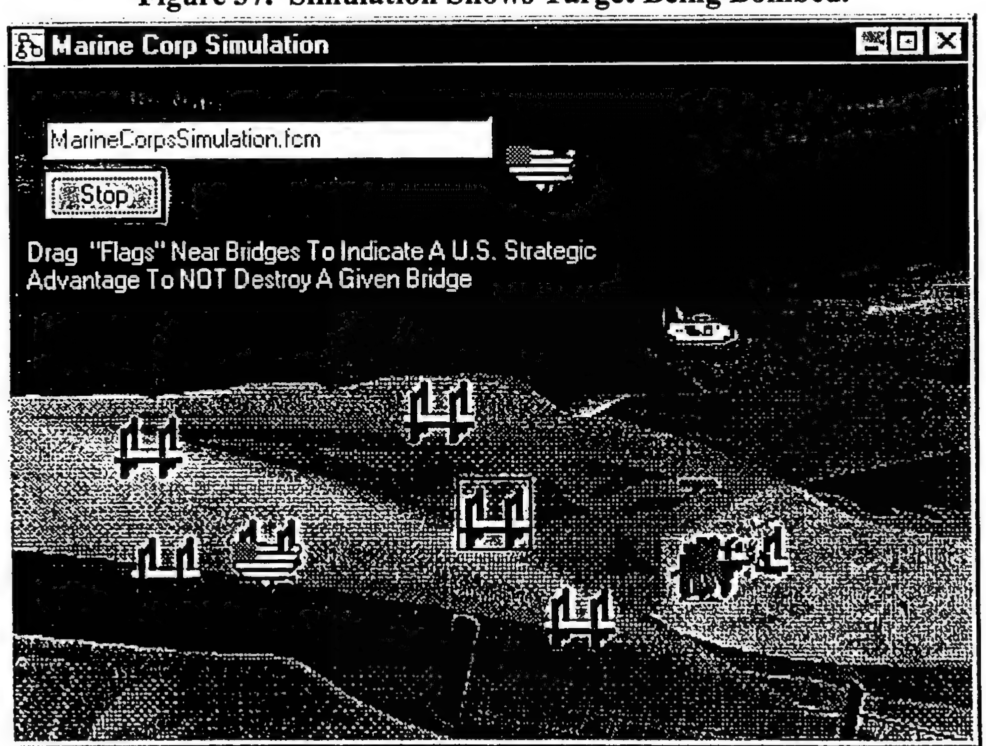


Figure 38. Simulation Shows Target Destroyed.

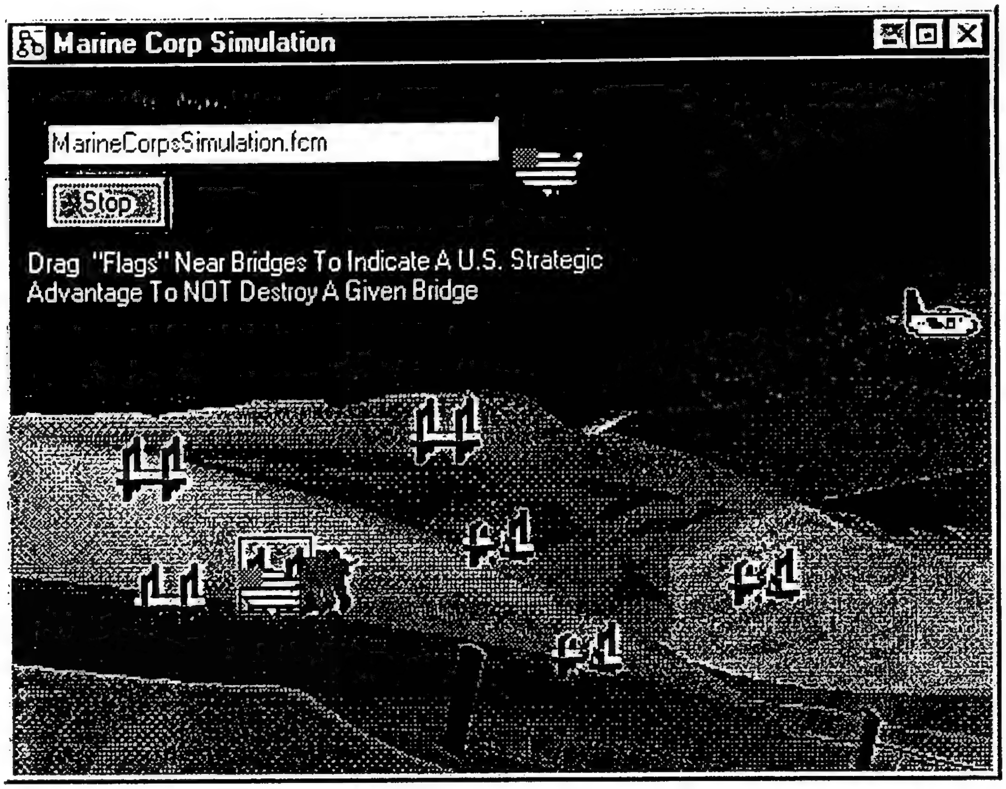


Figure 39. Target With Flag to Show U.S. Advantage.

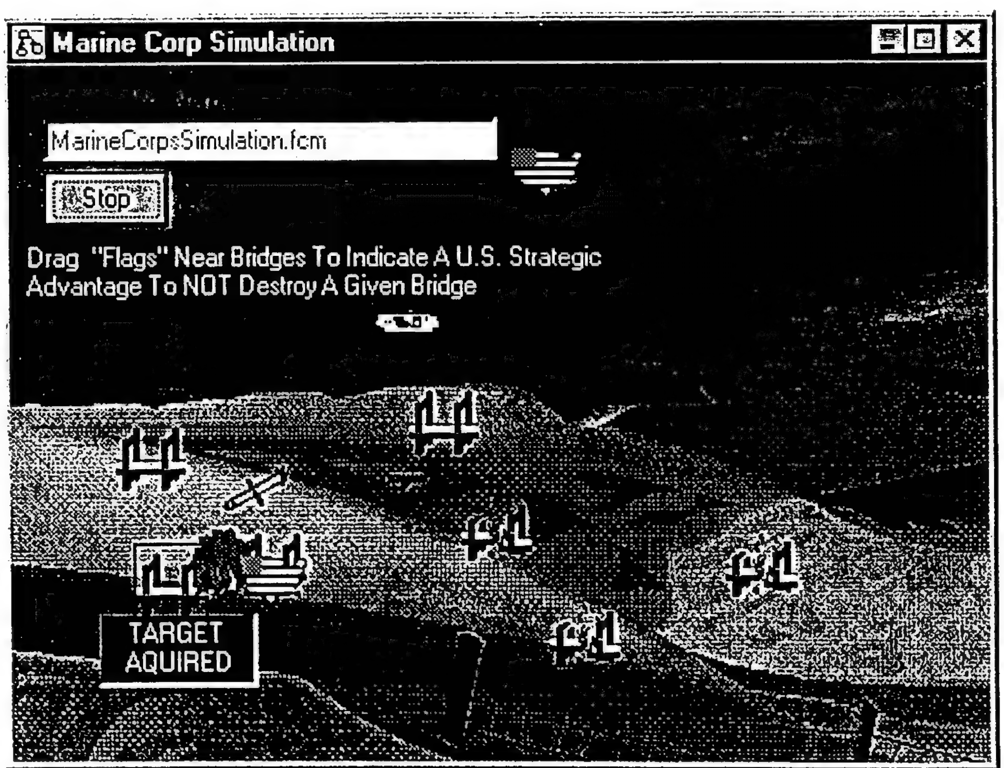


Figure 40. New Target With Large Value and Being Bombed.

III. TRANSITION PLAN, COMMERCIALIZATION, AND DUAL USE.

In order to transition this technology to the commercial market place and have a broad Marine Corps impact, TACAN can use the FCM approach for network management of diverse communications networks. Specific examples have been provided to the Marine Corps Systems Command. Other applications include chemical sensor data fusion, wargaming, political analysis, and general data fusion software.

IV. CONCLUSIONS

This program investigated a novel knowledge-based technology employed for command and control decisions in amphibious warfare. This technology is based upon the inherent causal inference and knowledge synthesis properties of the fuzzy cognitive maps which provide a nonlinear feedback dynamical system for modeling fuzzy causal knowledge process. Furthermore, fuzzy cognitive maps allow the synthesis of various different knowledge bases from different domain experts to generate a better and more robust knowledge representation. Based on the knowledge embedded in the predefined fuzzy cognitive maps, appropriate causal inference processes can be performed to enhance the decision making processes on various command and control systems. Investigations included the following: (1) methods to map the received information data into the node states of a fuzzy cognitive map, (2) algorithms to transfer the knowledge of various domain experts to the node links of a fuzzy cognitive map, and (3) causal inference rules and evolution algorithms for data discrimination and status prediction. With the results of these investigations, a prototype knowledge-based information fusion and data reasoning software module was created to demonstrate the concept of using fuzzy cognitive maps for reliable and command and control. There is a strong commercial and DoD need for this type of decision making technology, and TACAN has a strong prior record of commercialization of SBIR programs. The successful exploration and development of this knowledge-based system for reliable and robust decision making will eventually lead to a considerable increase in efficiency, reliability, and systems performance. Examples are areas such as complex manufacturing and quality control processes, information synthesis and analysis for medical or environmental and natural problems, and any applications that require integrated decisions and timely and accurate information.

V. PATENTS FILED, PENDING, OR PLANNED

None.

VI. MILITARY PERSONNEL OR ORGANIZATIONS BRIEFED

U.S. Navy SPAWAR (NCCOSC), San Diego, CA

U.S. Marine Corps Systems Command, Quantico, VA

VII. REFERENCES

[1] R. C. Luo, and M. G. Kay, "Information Integration and Fusion in Intelligent Systems," IEEE Trans. Systems, Man, and Cybernetics, Vol. 19, No. 5, pp. 901-931, 1989.

[2] O. D. Faugeras, N. Ayache, and B. Faverjon, "Building Visual Maps by Combining Noisy Stereo Measurements," in Proc. IEEE Conf. Robotics and Automat., San Francisco, CA, Apr. 1986, pp. 1433-1438.

[3] D. J. Kriegman, E. Triendl, and T. O. Binford, "A Mobile robot: Sensing, Planning and Locomotion," in Proc. IEEE Int. Conf. Robotics and Automat., Raleigh, NC, Mar. 1987, pp. 402-408.

[4] R. C. Luo, M. Lin, and R. S. Scherp, "Dynamic Multi-Sensor Data Fusion for Intelligent Robots," IEEE J. Robot. Automat., Vol. RA-4, No. 4, pp. 386-396, 1988.

[5] R. C. Luo, and M. Lin, "Robot Multi-Sensor Fusion and Integration: optimum Estimation of Fused Sensor Data," in Proc. IEEE Int. Conf. Robotics and Automat., Philadelphia, PA, Apr. 1988, pp. 1076-1081.

[6] H. F. Durrant-Whyte, "Consistent Integration and Propagation of Disparate Sensor Observations,' Int. J. Robot. Res., Vol. 6, No. 3, pp. 3-24, 1987.

[7] H. F. Durrant-Whyte, "Sensor Models and Multi-Sensor Integration," Int. J. Robot. Res., Vol. 7, No. 6, pp. 97-113, 1988.

[8] R. McKendall, and M. Mintz, "Robust Fusion of Location Information," in Proc. IEEE Int. Conf. Robotics and Automat., Philadelphia, PA, Apr. 1988, pp. 1239-1244.

[9] M. Zeytinoglu, and M. Mintz, "Robust Fixed Size Confidence Procedures for a Restricted Parameter Space," Ann. Statist., Vol. 16, No. 3, pp. 1241-1253, 1988.

[10] Zs. M. Kovács-V, R. Guerrieri and G. Baccarani, "Cooperative Classifiers for High Quality Hand Printed Character Recognition," in Proc. World Congress on Neural Networks (WCNN), Vol. I, 1993, pp. 186-189.

[11] A. DeCegama, and J. Smith, "Neural Networks and Genetic Algorithms for Combinatorial Optimization of Sensor Data Fusion," in Proc. SPIE, Vol. 1699, 1992, pp. 108-115.

[12] J. R. Brown, D. Bergondy, and S. Archer, "Comparison of Neural Network Classifiers to Quadratic Classifiers for Sensor Fusion," in Proc. SPIE, Vol. 1469, 1991, pp. 539-543.

[13] D. W. Ruck, S. K. Rogers, and M. Kabrisky, "information Fusion Classification with a Multi-layer Perceptron," in Proc. Int. Joint. Conf. Neural Networks (IJCNN), San Diego, Vol. II, 1990, pp. 863-868.

[14] B. Kosko, "Fuzzy Cognitive Maps," Int. J. Man-Machine Studies, Vol. 24, pp. 65-75, 1986.

[15] B. Kosko, "Adaptive Inference in Fuzzy Knowledge Networks," in Proc. First Int. Conf. Neural Networks, Vol. 2, 1987, pp. 261-268.

[16] W. R. Taber, and M. A. Siegel, "Estimation of Expert Weights Using Fuzzy Cognitive Maps," in Proc. First Int. Conf. Neural Networks, Vol. 2, 1987, pp. 319-325.

[17] W. Zhang, and S. Chen, "A Logical Architecture for Cognitive Maps," in Proc. 2nd Int. Conf. Neural Networks, Vol. 1, 1988, pp. 231-238.

[18] M. A. Styblinski, and B. D. Meyer, "Fuzzy Cognitive Maps, Signal Flow Graphs, and Qualitative Circuit Analysis," in Proc. Int. Joint Conf. Neural Networks (IJCNN), San Diego, Vol. II, 1988, pp. 549-556.

[19] M. Hagiwara, "Extended Fuzzy Cognitive Maps," in Proc. IEEE Int. Conf. Fuzzy Systems, 1992, pp. 795-801.

[20] R. Taber, "Coding the Fuzzy Cognitive Map as A Rule Base," in Proc. World Congress Neural Networks (WCNN), Vol. II, 1993, pp. 2-8.

[21] P. Craiger, "Modeling Dynamic Social and Psychological Processes with Fuzzy Cognitive Maps," in Proc. 3rd Conf. Fuzzy Systems, IEEE World Congress on Computational Intelligence, 1994, pp. 1873-1877.

[22] P. Craiger, "Causal Structure, Model Inferences, and Fuzzy Cognitive Maps: Help for the Behavioral Scientist," in Proc. World Congress Neural Networks (WCNN), Vol. I, 1994, pp. 836-841.

[23] J. A. Dickerson, B. Kosko, "Adaptive Fuzzy Cognitive Maps in Virtual Worlds," in Proc. World Congress Neural Networks (WCNN), Vol. IV, 1994, pp. 484-492.

[24] P. C. Silva, "Fuzzy Cognitive Maps Over Possible Worlds," in Proc. IEEE Int. Conf. Fuzzy Systems, Vol. II, pp. 555-560, 1995.

[25] R. S. Satur, Z-Q Liu, and M. Gahegan, "Multi-Layered FCMs Applied to Context Dependent Learning," in Proc. IEEE Int. Conf. Fuzzy Systems, Vol. II, pp. 561-568, 1995.

[26] P. C. Silva, "Fuzzy Cognitive Maps in Multi-Agent Environments," Topics in Artificial Intelligence, 4th Congress of the Italian Association for Artificial Intelligence, 1995, pp. 37-43.

[27] M. Schneider, E. Shnaider, A. Kandel, and G. Chew, "Constructing Fuzzy Cognitive Maps," in Proc. IEEE Int. Conf. Fuzzy Systems, Vol. IV, pp. 2281-2288, 1995.